

WEST**End of Result Set**Generate Collection Print

L4: Entry 3 of 3

File: USPT

Nov 22, 1994

DOCUMENT-IDENTIFIER: US 5367684 A

**** See image for Certificate of Correction ****

TITLE: Register allocation using an improved register candidate usage matrix

CLAIMS:

1. In a computer system comprising a plurality of instructions and registers, a computer implemented method for a register allocator of a code generator of a compiler of said computer system to allocate said registers to instructions being generated for a program being compiled by said compiler, said method comprising the steps of:

a) said register allocator building a collection of register candidate usage indication vectors for register candidates of said instructions being generated for said program being compiled, said instructions being divided in a plurality of basic blocks, each of said basic blocks comprising an entry, a body, and an exit, said register candidate usage indication vectors comprising indicators denoting at least whether said register candidates are live at said entries, in said bodies, and at said exits of said basic blocks, and denoting live ranges of said register candidates;

b) said register allocator allocating said registers to said instructions of each of said basic block, on a block by block basis, using said indicators of said collection of register candidate usage indication vectors to determine whether any two register candidates interfere with each other within a basic block; and

c) said register allocator merging said register allocation for each of said basic block, inserting spill code for each live range split of a register candidate, said split code insertions being made at corresponding points in said instructions being generated where said register candidate live ranges are split.

7. In a computer system comprising a plurality of instructions and registers, an improved register allocator of a code generator of a compiler for allocating said registers to instructions being generated for a program being compiled by said compiler, said improved register allocator comprising:

a) building means for building a collection of register candidate usage indication vectors for register candidates of said instructions being generated for said program being compiled, said instructions being divided in a plurality of basic blocks, each of said basic blocks comprising an entry, a body, and an exit, said register candidate usage indication vectors comprising indicators denoting at least whether said register candidates are live at said entries, in said bodies, and at said exits of said basic blocks, and denoting live ranges of said register candidates;

b) allocation means coupled to said building means for allocating said registers to said instructions of each of said basic block, on a block by block basis, using said indicators of said collection of register candidate usage indication vectors to determine whether any two register candidates interfere with each other within a basic block; and

c) merging means coupled to said allocation means for merging said register allocation for each of said basic block, inserting spill code for each live range split of a register candidate, said spill code insertions being made at

corresponding points in said instructions being generated where said register candidate live ranges are split.

13. In a computer system comprising a plurality of instructions and registers, a code generator of a compiler for generating instructions for a program being compiled, wherein said registers are allocated to said instructions being generated, said code generator comprising:

a) a translator for translating intermediate representations of said program into ordered instruction blocks;

b) an optimizer coupled to said translator for receiving said intermediate representations and ordered instruction blocks, and optimizing said instructions being generated for said program being compiled;

c) an improved register allocator coupled to said improved optimizer for receiving said optimized intermediate representations and ordered instruction blocks, and allocating said registers to said instructions of said ordered instruction blocks being generated, said improved register allocator comprising,

c.1) building means for building a collection of register candidate usage indication vectors for register candidates of said instructions being generated for said program being compiled, said instructions being divided in a plurality of basic blocks, each of said basic blocks comprising an entry, a body, and an exit, said register candidate usage indication vectors comprising indicators denoting at least whether said register candidates are live at said entries, in said bodies, and at said exits of said basic blocks, and denoting live ranges of said register candidates,

c.2) allocation means coupled to said building means for allocating said registers to said instructions of each of said basic block, on a block by block basis, using said indicators of said collection of register candidate usage indication vectors to determine whether any two register candidates interfere with each other within a basic block, and

c.3) merging means coupled to said allocation means for merging said register allocation for each of said basic block, inserting spill code for each live range split of a register candidate, said spill code insertions being made at corresponding points in said instructions being generated where said register candidate live ranges are split; and

d) a scheduler coupled to said register allocator for receiving said optimized and register allocated intermediate representations and scheduling said instructions being generated for said program being compiled; and

e) an assembly generator coupled to said scheduler for receiving said optimized, register allocated, and scheduled intermediate representations and generating said instructions for said program being compiled.

15. In a computer system comprising a plurality of instructions and registers, a compiler for compiling a program, wherein instructions being generated for said program being compiled having said registers allocated to them, said compiler comprising:

a) a parser for receiving source code of said program, parsing said source code and generating tokenized statements for said program;

b) an intermediate representation builder coupled to said parser for receiving said tokenized statements and building intermediate representations for said program; and

c) a code generator coupled to said intermediate representation builder for receiving said intermediate representations, and generating instructions for said program, said code generator comprising an improved register allocator for allocating said registers to said instructions being generated, said improved register allocator comprising,

c.1) building means for building a collection of register candidate usage indication vectors for register candidates of said instructions being generated for said

program being compiled, said instructions being divided in a plurality of basic blocks, each of said basic blocks comprising an entry, a body, and an exit, said register candidate usage indication vectors comprising indicators denoting at least whether said register candidates are live at said entries, in said bodies, and at said exits of said basic blocks, and denoting live ranges of said register candidates,

c.2) allocation means coupled to said building means for allocating said registers to said instructions of each of said basic block, on a block by block basis, using said indicators of said collection of register candidate usage indication vectors to determine whether any two register candidates interfere with each other within a basic block, and

c.3) merging means coupled to said allocation means for merging said register allocation for each of said basic block, inserting spill code for each live range split of a register candidate, said spill code insertions being made at corresponding points in said instructions being generated where said register candidate live ranges are split.

17. A computer system comprising:

a) a plurality of instructions;

b) a plurality of registers;

c) a compiler for compiling a program, wherein instructions being generated for said program being compiled having said registers allocated to them, said compiler comprising an improved register allocator for allocating said registers to said instructions being generated, said improved register allocator comprising,

c.1) building means for building a collection of register candidate usage indication vectors for register candidates of said instructions being generated for said program being compiled, said instructions being divided in a plurality of basic blocks, each of said basic blocks comprising an entry, a body, and an exit, said register candidate usage indication vectors comprising indicators denoting at least whether said register candidates are live at said entries, in said bodies, and at said exits of said basic blocks, and denoting live ranges of said register candidates,

c.2) allocation means coupled to said building means for allocating said registers to said instructions of each of said basic block, on a block by block basis, using said indicator of said collection of register candidate usage indication vectors to determine whether any two register candidates interfere with each other within a basic block, and

c.3) merging means coupled to said allocation means for merging said register allocation for each of said basic block, inserting spill code for each live range split of a register candidate, said spill code insertions being made at corresponding points in said instructions being generated where said register candidate live ranges are split.